

# A novel classification and clustering algorithms for intrusion detection system on convolutional neural network

Mathiyalagan Ramasamy<sup>1</sup>, Pamela Vinitha Eric<sup>2</sup>

<sup>1</sup>Department of CSE, New Horizon College of Engineering, Bengaluru, India

<sup>2</sup>Computer Science and Engineering, Presidency University, Bengaluru, India

## Article Info

### Article history:

Received May 28, 2022

Revised Jun 10, 2022

Accepted Jul 27, 2022

### Keywords:

Computer networks

Deep learning

Evolutionary techniques

Feature selection

Feed forward neural network

Intrusions detection system

Network security

## ABSTRACT

At present data transmission widely uses wireless network framework for transmitting large volume of data. It generates numerous security problems and privacy issues which laid a way for developing IDS. IDS act as preventive technique in securing computer networks. Previously there are numerous metaheuristic and deep learning algorithms used in IDS for detecting threats. Some are affected by dynamic growth of feature spaces and others are degraded in performance during detection of threats. One fine-grained model for intrusion detection can be developed by selecting accurate features and testing them with the intelligent algorithms. Based on these explorations, in this research IDS is implemented with intelligence from preprocessing to feature classification. At first stage, data preprocessing is done using binning concept to reduce noise. Secondly feature selection is done dynamically using dynamic tree growth algorithm with fire fly optimization techniques. Finally, these features are processed using DTB-FFNN for detecting anomalies perfectly. This DTB-FFNN is evaluated with popular KDD dataset. Our proposed model cable news network (CNN)-classification is compared with existing intelligent techniques: feed forward deep neural network, support vectors machines, decision tree, and CNN-clustering is compared with k-means, density-based spatial clustering of applications with noise (DBSCAN). The experimental outcome proves that dynamic tree based FFNN and CNN-clustering produce higher accuracy than the existing models.

This is an open access article under the [CC BY-SA](#) license.



## Corresponding Author:

Mathiyalagan Ramasamy

New Horizon College of Engineering

Bangaluru, India

E-mail: mathi.prajval@gmail.com

## 1. INTRODUCTION

Intrusion detection is a device system helps to monitor the traffic in the networks for detecting threatened activity when discovered in network. This device is designed using software application for monitoring harmful features entering in the network for infringe network policies. If any malicious data enters the network and trying to access, then it is detected by intrusions detection system (IDS). The malicious activity is reported to administrator [1]–[3]. Wireless computer networks are highly prone to numerous malicious attacks. Wireless networks are naturally opened with flexibility and mobility in medium [4], [5]. These networks are secured using intelligent intrusion detection system [6]. IDS can be classified in to five types based on host IDS, network IDS, protocol IDS, application protocol IDS and hybrid IDS [7]. Further above IDS system uses three types of detection techniques using signature IDS, anomaly IDS and hybrid IDS [8], [9]. IDS based on anomaly detection monitor and analysis the network as normal process and

detect intrusion deviations in network. IDS based on signature uses predefined keywords of malwares to pinpoint the intrusions in networks. Here databases are updated manually by system administrators.

IDS is considered as an effective detection technique in terms of malwares, unknown entry in network. Due to high performance of detective system, IDS achieves less false positive rates and high accuracy in classifying intruders' features [10]. The main crucial work is to analysis of machine learning techniques for IDS which helps to decrease false detection rate. When most of data processing environment communicates via wireless networks, we in this research design the intrusion detection using well defined deep learning techniques for wireless networks. IDS main role is to monitor the network with eagle eyes to detect the malwares entering to network for hacking data. our techniques must work like eagle eyes to detect intruders with high accuracy. The network intrusion detection system (NIDS) is capable of detecting malware entry in whole network through traffic feature. There is no need of all features for detecting the intruders in the network. When we choose consequential feature, it helps to minimize the execution time and increase malware detection rate. These weighted features favor the deep learning algorithms by increasing the accuracy of learning features. Major advantages of using feature selection and deep learning techniques are as follows: i) over fitting problem is prevented; ii) it avoids noise resistance; iii) executes detection operation in less time; iv) performance of prediction is improved.

Some of popular machine learning techniques in IDS are k-nearest neighbor (KNN), decision tree, support vector machine (SVM), random forest, naïve bayes and multi-layered perceptions operates with deep learning algorithms. Huge data in the IDS degrades the performance of the machine learning techniques. One of imperative method to overcome this problem is choosing suitable algorithms and classification model to improve efficiency of the IDS system. This research tries to improve deep learning strategy with good feature selection models. Deep learning was first introduced by LeCun *et al.* [11] as advanced machine learning methodology for complex data computation. Multiple machine learning structures are represented and discussed in [12]. Advanced fields like language processing, image processing and medical research are more success by using deep learning techniques [13]–[15]. This makes deep learning to include in IDS classification models. The main contribution of this article includes,

- a. Reducing the feature noises with high accuracy using normalized PCA. Then the feature selection is processed by tree growth algorithm with fire fly optimizing techniques.
- b. Selected features are deeply analyzed using the feed forward neural network to ensure the selected feature is outsider or intruder in the network.
- c. The accuracy of prediction and less false positive rate is obtained by this proposed technology.

Rest of the paper is organized as follows: section 2 studies the existing articles on intrusion detection; section 3 describes the proposed methodology and working principal; section 4 describes result and experimental setup; section 5 concludes the article with future scope of the article.

## 2. LITERATURE SURVEY

In this study we tend to discuss various feature selection models in feature selection as well as classification using many good algorithms in AI. Deep leaning algorithm with feature selection provides optimal solution in various applications. The concept of IDS using non-symmetric deep auto-encoder (NDAE) and for the feature classification by using stacked based NDAE provides good accuracy. This IDS concept evaluates by using NSL-KDD and knowledge discovery in database (KDD) cup 99 datasets. The performance of two datasets yielded an accuracy of 97:85% on the KDD Cup 99 dataset than accuracy of NSL-KDD dataset [16]–[22].

Mondal *et al.* [23] proposed feature selection process based on the labeled multi-objective algorithm using the concept of mutual information (MOMI). This MOMI evaluate the experiments using WEKA tool in the classifiers of SVM and naive bayes (NB) [24]. For the feature selection algorithm this paper proposed a framework of multilayer perceptron using controlled redundancy (FSMLP-CoR) [25]. In this approach of FSMLP-CoR architecture has input layer, output layer with multiple hidden layers and it is used for regression, classification and prediction in several domains [26]–[28]. In the dataset KDD cup 99 for feature selection ant colony optimization (ACO) algorithm is used in intrusion detection. This dataset has 41 features and ACO technique implements how ants remember their path using pheromones. It also evaluates the classifier library of binary SVM in WEKA [29]. Hybrid model of SVM classifier with genetic algorithm in the intrusion detection system. The advantage of this hybrid model can reduce the features and implement it using 10 features [30]. In reducing the number of features, it will increase the detection rate and performance of network intrusion detection.

Several NIDS models is based on metaheuristic algorithms of particle swarm optimization (PSO) [31]–[33], firefly optimization algorithm (FFA) [34], [35], genetic algorithm (GA) [36]–[38] and grey wolf

optimizer (GWO) [39]–[41] for optimal detection of intruders in the network. Table 1 describes the survey of various feature selection model.

Table 1. Survey on feature selection in IDS

Author name	Feature selection	Dataset	Obtained features
Al-Yaseen (2019) [42]	Firefly algorithm and SVM	NSL-KDD	F1, f2, f3, f9, f8, f6, f11, f12, f11, f13, f19, f16, f28, f26, f34, f32, f31, f37, f35, f43, f41.
Aljawameh <i>et al.</i> (2019) [43]	J48 (C4.5 decision tree) classifier	NSL-KDD	F1-f9, f11-f13, f16, f19, f17, f22-f26, f28, f31, f32, f35, f34, f37, f40, f39, f41, f43
Taherkhani <i>et al.</i> (2018) [44]	deep boltzmann machines	NSL-KDD	F1-f9, f11-f13, f16, f19, f17, f22-f26, f28, f31, f32, f34, f37, f40, f39, f41, f43
Ahmad <i>et al.</i> (2018) [45]	SVM, extreme learning machine (ELM) and random forest (RF)	NSL-KDD	F1-f9, f11-f13, f16, f19, f17, f22-f26, f28, f31, f32, f35, f34, f37, f40, f39, f41, f43
Farahnakian <i>et al.</i> (2018) [46]	deep auto-encoder based	NSL-KDD and KDD Cup 99	F1-f9, f11-f13, f16, f19, f17, f22-f26, f28, f31, f32, f35, f34, f37, f40, f39, f41, f43
Yin <i>et al.</i> (2017) [47]	recurrent neural networks	NSL-KDD and KDD Cup 99	F1-f9, f11-f13, f16, f19, f17, f22-f26, f28, f31, f32, f35, f34, f37, f40, f39, f41, f43
Khammassi and Krichen (2017) [48]	GA-LR wrapper approach	KDD Cup 99	F1-f9, f16, f11, f24, f25, f21, f27, f28, f32, f35-f37, f41, f42, f39
Mehmod <i>et al.</i> (2016) [49]	ant colony optimization (ACO)	KDD Cup 99	F1-f9, f11-f13, f16, f19, f17, f22-f26, f28, f31, f32, f35, f34, f37, f40, f39, f41, f43
Agarwal <i>et al.</i> (2016) [50]	flooding DoS attacks	KDD Cup 99	F1-f9, f11-f13, f16, f19, f17, f22-f26, f28, f31, f32, f35, f34, f37, f40, f39, f41, f43

Wireless communication network plays a vital role in data generation and transmission with safe transmission. It is popular due its wireless infrastructure. Some networks are wireless local area network (W-LAN), this LAN is part of the family IEEE 802.11 used in several industrial applications and smart home building applications. Wi-Fi based protected technique WAP is used to provide security form DOS attacks, brute force attack, and network discovery problems [51]–[54]. This research uses wired and wireless IDS review using machine and deep learning strategy.

### 3. PROPOSED METHOD FOR IDS

Network intrusion detection system is an important member of networks for detecting and stopping anonyms entry in to the network. while accessing the network, system features with personal features are identified by routers and networking devices. Every entry is monitored by the IDS device and if any intrusion feature is detected, it will inform to administration of network system. In this research we implement IDS in three phases: Figure 1 shows general working architecture of our proposed model. In first phase data is preprocessed by removing all high-level noises. In second phase tree growth-based firefly algorithm is used to detect the anonym's features. In third phase we use SVM and FFNN classifier for classifying the intrusions from selected features.

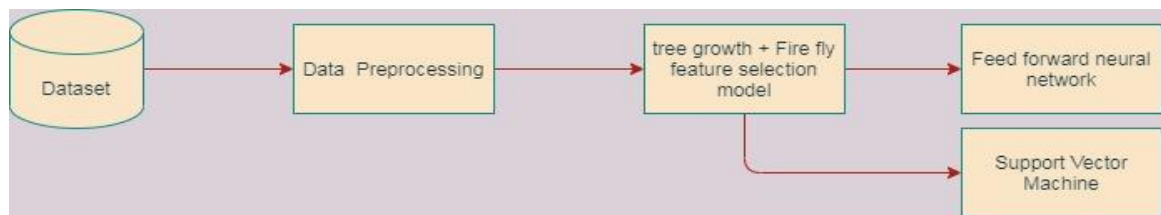


Figure 1. General architecture proposed model

#### 3.1. Dynamic tree-based feed forward neural network

##### 3.1.1. Basic tree growth ideology

Tree growth (TG) is swarm intelligence-based metaheuristic approach inspired by trees growing behavior in jungle [3]. Initially tree growth algorithm uses candidate set for constructing the tree. Then the tree population is divided using the fitness value. Best fitness tree is allocated in first group where it will grow further. In second group trees competitively grows with competition and moves to nearest of best trees with various angle to receive light. In third group replaces the weakest tree with new best one and fourth group where best tree multiplies to reproduce new one. Basically, tree growth algorithm has four mathematical backgrounds as follows:

**Steps to basic TG algorithm:**

**Step 1:** tree population is randomly generated as T1 initially. Fitness value used to select best first group tree using (1)

$$TR_i^{j+1} = \frac{TR_i^j}{\theta} + \text{ran}TR_i^j \quad (1)$$

Where i represent population among tree,  $\theta$  represents power reduction due to factors like aging, growth and reduced light (food) etc, 'ran' represents random tree selection parameter. Random numbers are binary 0,1 and root moves to  $\text{ran}TR_i^j$  growth rate. If new tree attains better value than the existing, then new is replaced for old tree. Tuning the  $\theta$  is core part of TGA.

**Step 2:** next tree T2 moves to best tree its various value of angle. Each tree in at N2 set uses (2) and (9) for computing the distance between the trees.

$$dis_i = \left( \sum_{i=1}^{T1+T2} (TR_{T2}^j - TR_i^j)^2 \right)^{\frac{1}{2}} \quad (2)$$

$$dis_i = \begin{cases} dis_i & \text{if } TR_{T2}^j \neq TR_i^j \\ \infty & \text{if } TR_{T2}^j = TR_i^j \end{cases} \quad (3)$$

Where t2 represents current tree in the 'i' th population. Two outcome x1 and x2 are chosen within minimum distance dis (i) by using (4)

$$y = \Delta x_1 + (1 - \Delta)x_2 \quad (4)$$

**Step 3:** tree which are worst T3 are eliminated. The size of population is,

$$T = T1 + T2 + T3 \quad (5)$$

**Step 4:** masking operation is performed and T4 is generated.

**3.1.2. Tuning parameter**

The factor  $\theta$  is important parameter used to adjust and select the best of tree in the tree population. In literature paper [3] used to tune the  $\theta$  before the tree simulation. Also, in literature [3] it is tuned after simulation. These two scenarios tuned in extreme conditions. It seems this tuning cannot provide reasonable tree growth strategy. In this article we claim to tune  $\theta$  from to ending at each iteration. As discussed in TGA,  $\theta$  is defined as reduction power of tree. Growth rate of tree cannot be constant at any time. In jungle, tree growth rate is based on soil nutrients. Based on this slow growing strategy we tune  $\theta$  linearly based on increasing pattern using (6).

$$\theta = 0.75 * \left( 1 + \frac{3c}{\text{Total(iter)}} \right) \quad (6)$$

where, c is current iteration, total (iter) denotes total number of iteration tree algorithm computes. This linear method increases tree reduction power with it age and nutrient. The result of the proposed ideology improves the feature selection model.

**3.1.3. Embedding TGA with fire fly strategy**

Fire fly is the swarm intelligence based meta-heuristic search tool used to search an optimal feature among the large feature in our research article. This firefly (FF) algorithm is based on social behavior of flies with lighting mathematical model [3]. FF algorithm works by using light intensity variation and attractiveness. For optimal selection of feature, objective function is designed. Firefly light intensity (I) is proportional to objective function. Contraction of light by flies using Gaussian distance procedure is given in [3] and based on these ideology and attractiveness parameter, new parameter is defined in enhanced tree growth fire fly (TGFF) algorithm

The best initial solution cannot always produce best final output [36]. Because, after each iteration features may be best or better or good. This makes of exploration and exploitation imbalanced in most of metaheuristic techniques. Proposed enhanced TGFF algorithm is given in algorithm 1.

**Algorithm 1: TGFF****Input:** Features from dataset**Output:** Selected featuresParameter initialization:  $\lambda$ ,  $\Theta$ ,  $P_1, P_2$ ;

Initial population of tree TR is generated;

Compute population in descending order;

Light intensity  $L_i$  of tree  $TR_i$  is defined  $g(TR)$ ;absorption of light  $\gamma$ ; quasi reflection solution and quasi reflection population is defined  $TR^q$ While  $tr < \max\text{-iter}$ Update  $\Theta$  using equ (4);For  $i=1$  to  $P_1$  then doUpdate  $TR_i$  using equ (1);

Modify 0 or 1 using equ (6);

Fitness value is calculated (3);

If fitness value is better than previous feature,

Then update;

End;

For  $i = (P_1+1)$  to  $(P_1 + P_2)$  computeDistance  $dis(i)$  is calculated using equ (2) and (3);Select  $x1=TR(dis_1)$  and  $x2=TR(dis_2)$ ;Sort  $dis_i$ ;Calculate the  $y$  using equ (4);

Update tree TR (i) using equ (5);

Change TR as 0 or 1 using equ (1)

If fitness value is better than previous feature,

Then update;

End;

For  $i = (P_2+1)$  to  $P$  do

Randomly compute the tree TR (i);

End

For  $k = 1$  to  $P_4$  doInitialize random  $r_k$ ;Choose solution  $TR_f$  from  $P_1$ ;Perform mask operation [2] for  $r_k$  and  $TR_f$ ;

End;

// Embedding firefly feature selection in binary TGA//

Compute union operation of  $\cup TR^q$ ;

Based on fitness value sort all features;

Select best features in the TR;

While  $f < \max\text{-iter}$  do $i = 1$  to  $n$  do $j = 1$  to  $i$  do $L_j > L_i$  then move  $j$  to  $i$  direction;Attractiveness is changed with distance  $dis$  as  $\exp[-\gamma dis]$ Apply the function sigmoid  $\text{sig}(x)$  asIf  $\text{sig}(x) < 0.5$  then feature is set as 0;

Else

Feature is set as 1;

End if ;

New solution is evaluated and old worst solution is deleted;

Light intensity updated;

End if

End for

End for

Now all firefly ranked in order and best feature is selected among them

End while;

### 3.2. Convolutional neural network-based clustering

Figure 1 shows the technical architecture proposed in this research for KDD based on CNN (KDD CNN), which consists of two parts: a similarity measuring technique and a two-stage clustering process. In the similarity measurement procedure, similarity between data is processed by training KDD in CNN. KDD dataset is divided into four categories based on similarity which in turn used as the training set for CNN in two-step clustering. This process is known as preliminary clustering. Then, in the target clustering phase, CNN is trained using the preliminary clustering findings, and the test set is separated into three groups. These two sections will be described in full further down.

#### 3.2.1. Similarity measurement algorithm

Figure 2 depicts the construction of CNN. The convolution portion is made up of three-layer convolution blocks, each with sizes of 256, 512, and 256. Each block is made up of a convolution layer filled with 1-D kernels without striding, followed by a BN layer, which is subsequently activated by the ReLU function [17]. The block sizes in the kernels are 16; 10; 6. The convolution layer is connected to the global average pooling layer. The linear layer provides the output of the similarity measurement.

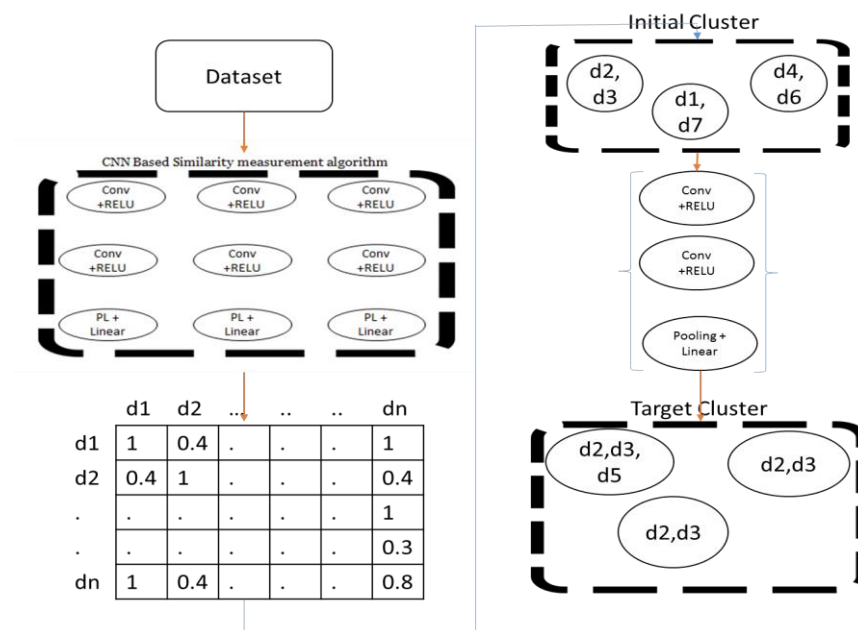


Figure 2. CNN based clustering

#### 3.2.2. Two-stage clustering

The suggested two-step clustering algorithm consists of the following two steps. In the initial step, the cluster construction approach is utilized to combine partial data in the dataset with a higher similarity. The cluster generation algorithm. The cluster construction process is dependent on the data selection criteria, or which data should be clustered first. The value of similarity and the similarity rating with the current data are utilized to judge in this algorithm. If you solely utilize the similarity value to determine whether or not to cluster, you may end up collecting too much data, especially if the similarity difference between data is small, lengthening your training time. The collection of data with low similarity may occur from selecting only a small quantity of data based on similarity rating, limiting clustering accuracy. The clusters created in the first stage are used as the network's training set, and the KDD dataset is divided as the network's test set in the second step. The construction of CNN in the two-step clustering phase is the same as in Algorithm 1, with the exception that the last layer is Softmax rather than liner.

## 4. RESULTS AND EXPERIMENTS

### 4.1. Dataset description

In this research work we tend to use dataset called UNSW-NB15. This dataset is developed for network security. It is a hybrid dataset which contains current real data to attacked data. The tool perfect

storm IXIA used to create dataset. It is composed of nine families of attacks and real data. It consists of 49 features in the network flow. Most of the features are listed in table. Our proposed result is compared with various feature selection model in [3], [4] and result is discussed. Features are listed in Table 2.

Table 2. Dataset features

Feature id	Feature name	Feature id	Feature name	Feature id	Feature name	Feature id	Feature name
1	State	13	dBytes	25	Synack	37	ct_dst_ltm
2	Service	14	Sbytes	26	Tcprrt	38	ct_dst_src_ltm
3	ID	15	Sloss	27	Ackdat	39	ct_ftp_cmd
4	dur	16	Dloss	28	Dmean	40	ct_flw_http_mthd
5	sload	17	Dinpkt	29	Smean	41	ct_srv_dst
6	dload	18	Sinpkt	30	Transdepth	42	ct_src_ltm
7	proto	19	Djit	31	ct_srv_src	43	attack_cat
8	spkts	20	Sjit	32	Response_body-len	44	label
9	dpkts	21	Stcpt	33	ct_state_ttl	45	is_sm_ips_ports
10	rate	22	Dtcpt	34	ct_src_deport_ltm		
11	dttl	23	Swin	35	ct_dst_deport_ltm		
12	Sttl	24	Dwin	36	is_ftp_login		

#### 4.2. Result evaluation

The proposed efficiency is accessed with several model and performance metrics. The metrics we used are true positive, true negative, false positive, false negative [3]. The confusion matrix is displayed in Table 3. It calculates true positive rates (TPR), true negative rates (TNR), false negative rates (FNR) and false positive rates (FPR). Based on the values of following metrics, factors like precision, f-measure, sensitivity and accuracy are calculated and comparison graph is given. confusion matrix is discussed in Table 3.

Table 3. Confusion matrix

		Prediction	
		Real	Intrusion
Actual	Real	True positive (P)	False negative (Q)
	Intrusion	False positive (R)	True negative (S)

#### 4.3. Result and discussion

##### 4.3.1. Selected IDS feature experimental evaluation:

In this experiment, results are computed using Windows 10 operating system, 6.0 GB RAM, 3.40 GHZ, i7 CPU. The open-source anaconda python is used for experimental execution [3]. The Table 4 lists the selected features that the proposed model is detected as KDDK intrusion in network.

Table 4. Number of selected features

Methods	Selected features	Number of features
PSO	F2, f4, f5, f7, f11, f16, f12, f17, f19, f18, f20, f23, f22, f24, f26, f27, f25, f30	25
FFA	F33, f31, f39, f34, f43, f41, f40	21
GWO	F1, f2, f3, f9, f8, f6, f11, f12, f10, f12, f11, f13, f19, f16, f28, f26, f34, f32, f31, f37, f35, f43, f41	20
GA	F1-f9, f16, f11, f24, f25, f21, f27, f28, f32, f35-f37, f41-f42, f39	23
PSO+GWO+FFA+GA	F1-f9, f11-f13, f16, f19, f17, f22-f26, f28, f31, f32, f35, f34, f37, f40.f39, f41, f43	30
Proposed TGFFA	F1-f9, f11-f16, f19, f20, f22-f30, f31, f33, f35-f39, f40, f41, f43	36

In this Table 4, feature that are considered as intrusion are detected by various feature optimizer and selection algorithm. Although literature work [3] detects most of attacks, still it lags in prediction accuracy. Our proposed model detects most of the attacks and improves the accuracy of IDS system.

##### 4.3.2. Experimental evaluation

Our proposed model is evaluated based on two classifiers: SVM and FFDNN. Other two classifier does not provide best result among the four. K-NN algorithm produce 97.32% of accuracy in detecting the

intrusion, whereas NB achieves only 97% in detecting the intrusion. Now we discuss result in detail of TGFF-SVM and TGFF –FFDNN. Table 5 presents various results of existing techniques.

Table 5. The result obtained using SVM classifier

Method	TPR (%)	TNR (%)	FPR (%)	FNR (%)
PSO	79.86	97.32	8.72	3.66
GWO	92.34	78.61	26.43	5.21
GA	96.95	77.23	18.16	3.89
FFA	93.56	77.39	28.75	4.77
PSO+GWO+GA+FFA	96.87	83.65	15.66	2.89
PROPOSED TGFF	97.12	97.54	12.18	2.23

The Figure 3 presents the outcome of SVM classifier with various IDS feature detection methodologies. From Figure 3, sensitivity of the detection is detected. It's clear that PSO and our proposed technique TGFFA provides best result. Also, PSO performs slight better in this process. Figure 3 depicts precision of the various detection algorithm. Proposed TGFFA shows high precision in detecting IDS. Figure 3 shows accuracy of IDS detection. The proposed TGFF performs better and meet good accuracy level when compared with other methods. in the F-measure analysis of Figure 3, proposed TGFF and PSO+GW+GA+FA performs moreover equally in predicting the real true feature and real wrong features in IDS. Table 6 discusses various results of proposed FFNN with TGFF algorithm.

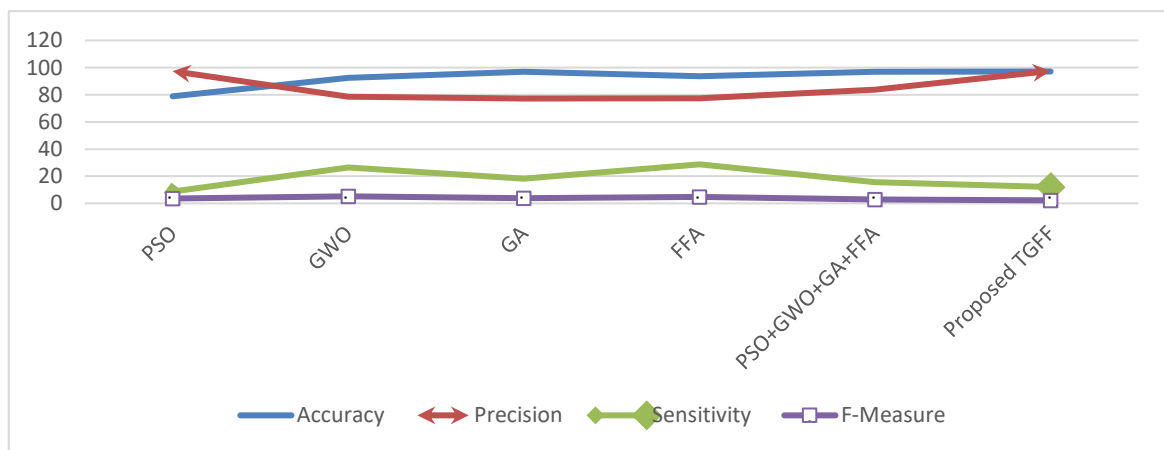


Figure 3. SVM classifier-based result evaluation with multiple methods

Table 6. The result obtained using FFDNN classifier

Method	TPR (%)	TNR (%)	FPR (%)	FNR (%)
PSO	80.86	97.98	7.12	3.16
GWO	93.34	78.11	23.53	4.87
GA	96.32	79.13	17.86	2.99
FFA	94.56	78.09	26.37	3.17
PSO+GWO+GA+FFA	96.32	84.33	14.66	3.59
PROPOSED TGFF	98.21	97.80	10.18	1.93

The Figure 3 presents the outcome of FDNN classifier with various IDS feature detection methodologies. From Figure 4, sensitivity of the detection is detected. It's clear that GWO, PSO and our proposed technique TGFFA provides best result. Also proposed performs slight better in this process. Figure 3 depicts precision of the various detection algorithm. Proposed TGFFA shows high precision in detecting IDS. Figure 3 shows accuracy of IDS detection. The proposed TGFF performs better and meet good accuracy level when compared with other methods. in the F-measure analysis of Figure 4, proposed TGFF performs moreover equally in predicting the real true feature and real wrong features in IDS.



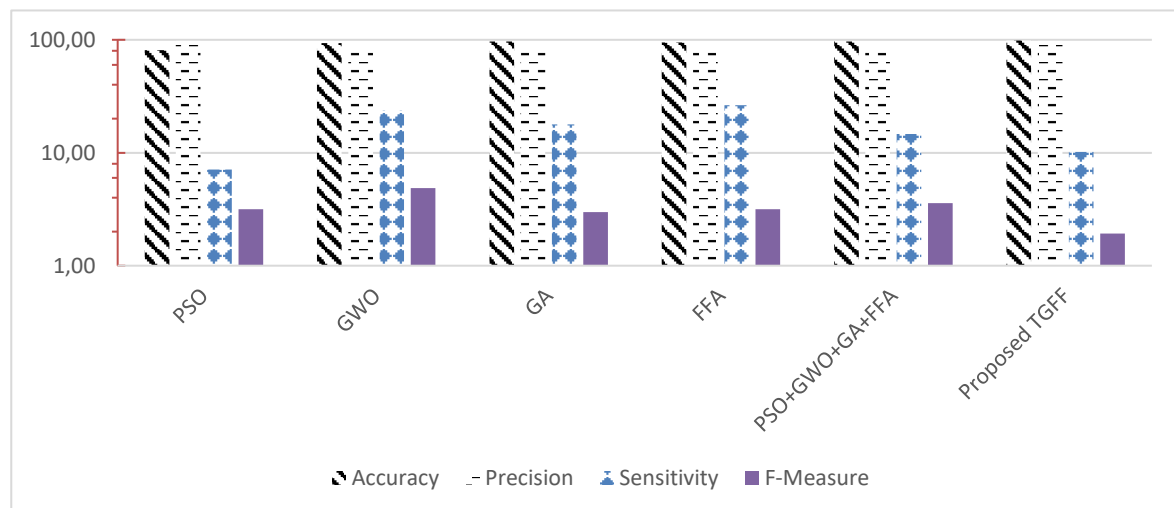


Figure 4. FDNN classifier-based result evaluation with multiple methods

## 5. CONCLUSION

Detecting the intrusions presents in the networks is a challenging research work. there are many features available in by the intrusion for acting as a real user in the network. Number of features affect the performance of detection in IDS system. The key role of our research work is choosing a best feature selection technique for tackling intelligent inntrusion avails in the network. In this IDS we use UNSW-NB15 dataset .Our proposed model using tree growth based firefly algorithm selects the feature efficiently than the other methods. The selected features are processed using feed forward neuarl network and classification of IDS is analysed. We are using TPR, TNR, FPR, FNR for performance analysis. In all evaluation result our proposed acheives 95.65%. Also PSO selects only 26 features from the dataset, GA selects 23 features and proposed TGFFA achieves 36 features with high efficiency and accuracy. In future artificial intelligent with blockchain technology can be used in monitoring the network intrusion.

## REFERENCES





- [1] S. M. Kasongo and Y. Sun, "A Deep Learning Method With Filter Based Feature Engineering for Wireless Intrusion Detection System," in *IEEE Access*, vol. 7, pp. 38597-38607, 2019, doi: 10.1109/ACCESS.2019.2905633.
- [2] T. Bezdan, D. Cvetnic, L. Gajic, M. Zivkovic, I. Strumberger, and N. Bacanin, "Feature selection by firefly algorithm with improved initialization strategy," *ECBS 2021: 7th Conference on the Engineering of Computer Based Systems*, 2021, pp. 1-8, doi: 10.1145/3459960.3459974.
- [3] C. Zhong, Y. Chen, and J. Peng, "Feature selection based on a novel improved tree growth algorithm," *International Journal of Computational Intelligence Systems*, vol. 13, no. 1, pp. 247-258, 2020, doi: 10.2991/ijcis.d.200219.001.
- [4] M. E. Aminanto, R. Choi, H. C. Tanuwidjaja, P. D. Yoo and K. Kim, "Deep Abstraction and Weighted Feature Selection for Wi-Fi Impersonation Detection," in *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 3, pp. 621-636, March 2018, doi: 10.1109/TIFS.2017.2762828.
- [5] C. Koliass, G. Kambourakis, A. Stavrou and S. Gritzalis, "Intrusion Detection in 802.11 Networks: Empirical Evaluation of Threats and a Public Dataset," in *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 184-208, Firstquarter 2016, doi: 10.1109/COMST.2015.2402161.
- [6] R. Mitchell and I.-R. Chen, "A survey of intrusion detection in wireless network applications," *Computer Communications*, vol. 42, no. 3, pp. 1-23, Apr. 2014. doi: 10.1016/j.comcom.2014.01.012.
- [7] J. Hu, X. Yu, D. Qiu and H. -H. Chen, "A simple and efficient hidden Markov model scheme for host-based anomaly intrusion detection," in *IEEE Network*, vol. 23, no. 1, pp. 42-47, January-February 2009, doi: 10.1109/MNET.2009.4804323.
- [8] D. A. Effendy, K. Kusriani and S. Sudarmawan, "Classification of intrusion detection system (IDS) based on computer network," *2017 2nd International conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, 2017, pp. 90-94, doi: 10.1109/ICITISEE.2017.8285566.
- [9] E. Viegas, A. O. Santin, A. França, R. Jasinski, V. A. Pedroni and L. S. Oliveira, "Towards an Energy-Efficient Anomaly-Based Intrusion Detection Engine for Embedded Systems," in *IEEE Transactions on Computers*, vol. 66, no. 1, pp. 163-177, 1 Jan. 2017, doi: 10.1109/TC.2016.2560839.
- [10] S. M. H. Bamakan, B. Amiri, M. Mirzabagheri, and Y. Shi, "A new intrusion detection approach using PSO based multiple criteria linear programming," *Procedia Computer Science*, vol. 55, pp. 231-237, Aug. 2015, doi: 10.1016/j.procs.2015.07.040.
- [11] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436-444, May 2015, doi: 10.1038/nature14539.
- [12] N. Shone, T. N. Ngoc, V. D. Phai and Q. Shi, "A Deep Learning Approach to Network Intrusion Detection," in *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 1, pp. 41-50, Feb. 2018, doi: 10.1109/TETCI.2017.2772792.

- [13] I. Lopez-Moreno, J. Gonzalez-Dominguez, D. Martinez, O. Plchot, J. Gonzalez-Rodriguez, and P. J. Moreno, "On the use of deep feedforward neural networks for automatic language identification," *Computer Speech & Language*, vol. 40, pp. 46-59, Nov. 2016, doi: 10.1016/j.csl.2016.03.001.
- [14] K. He, X. Zhang, S. Ren and J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1026-1034, doi: 10.1109/ICCV.2015.123.
- [15] S. Agatonovic-Kustrin and R. Beresford, "Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research," *Journal of Pharmaceutical and Biomedical Analysis*, vol. 22, no. 5, pp. 717-727, 2000, doi: 10.1016/S0731-7085(99)00272-1.
- [16] Y. Xin *et al.*, "Machine Learning and Deep Learning Methods for Cybersecurity," in *IEEE Access*, vol. 6, pp. 35365-35381, 2018, doi: 10.1109/ACCESS.2018.2836950.
- [17] H. Liu and L. Yu, "Toward integrating feature selection algorithms for classification and clustering," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 4, pp. 491-502, April 2005, doi: 10.1109/TKDE.2005.66.
- [18] S. S. Kannan and N. Ramaraj, "A novel hybrid feature selection via symmetrical uncertainty ranking based local memetic search algorithm," *Knowledge-Based Systems*, vol. 23, no. 6, pp. 580-585, Aug. 2010, doi: 10.1016/j.knsys.2010.03.016.
- [19] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, "The WEKA work bench," in *Data Mining: practical machine learning tools and techniques*, 4th ed. Burlington, MA, USA: Appendix, 2017, pp. 553-571.
- [20] R. Chakraborty and N. R. Pal, "Feature Selection Using a Neural Framework With Controlled Redundancy," in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 1, pp. 35-50, Jan. 2015, doi: 10.1109/TNNLS.2014.2308902.
- [21] L. Vanneschi and M. Castelli, "Multilayer perceptrons," In S. Ranganathan, M. Gribskov, K. Nakai, & C. Schönbach (Eds.), *Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics*, vol 1-3, pp. 612-620, doi: 10.1016/B978-0-12-809633-8.20339-7.
- [22] F. Murtagh, "Multilayer perceptrons for classification and regression," *Neurocomputing*, vol. 2, no. 5-6, pp. 183-197, Jul. 1991, doi: 10.1016/0925-2312(91)90023-5.
- [23] A. Mondal, A. Ghosh, and S. Ghosh, "Scaled and oriented object tracking using ensemble of multilayer perceptrons," *Applied Soft Computing*, vol. 73, pp. 1081-1094, Dec. 2018, doi: 10.1016/j.asoc.2018.09.028.
- [24] B. M. Aslahi-Shahri, R. Rahmani, M. Chizari, A. Maralani, M. Eslami, M. J. Golkar, and A. Ebrahimi, "A hybrid method consisting of GA and SVM for intrusion detection system," *Neural computing and applications*, vol. 27, pp. 1669-1676, 2016, doi: 10.1007/s00521-015-1964-2.
- [26] F. Marini and B. Walczak, "Particle swarm optimization (PSO)," *Chemometrics and Intelligent Laboratory Systems*, vol. 149, part B, 153-165, 2015, doi: 10.1016/j.chemolab.2015.08.020.
- [27] S. Srinoy, "Intrusion Detection Model Based On Particle Swarm Optimization and Support Vector Machine," *2007 IEEE Symposium on Computational Intelligence in Security and Defense Applications*, 2007, pp. 186-192, doi: 10.1109/CISDA.2007.368152.
- [28] M. Labani, P. Moradi, M. Jalili and X. Yu, "An Evolutionary Based Multi-Objective Filter Approach for Feature Selection," *2017 World Congress on Computing and Communication Technologies (WCCCT)*, 2017, pp. 151-154, doi: 10.1109/WCCCT.2016.44.
- [30] J. Kennedy and R. Eberhart, "Particle swarm optimization," *Proceedings of ICNN'95 - International Conference on Neural Networks*, 1995, pp. 1942-1948 vol.4, doi: 10.1109/ICNN.1995.488968.
- [31] X. S. Yang and X. He, "Firefly algorithm: Recent advances and applications," *International Journal of Swarm Intelligence*, vol. 1, no. 1, pp. 36-50, 2013, doi: 10.1504/IJSI.2013.055801.
- [32] B. Selvakumar and K. Muneeswaran, "Firefly algorithm-based feature selection for network intrusion detection," *Computers & Security*, vol. 81, pp. 148-155, 2019, doi: 10.1016/j.cose.2018.11.005.
- [33] M. Mitchell, *An introduction to genetic algorithms*, MIT press: Cambridge, MA, USA, 1998.
- [34] H. Gharraee and H. Hosseinvand, "A new feature selection IDS based on genetic algorithm and SVM," *2016 8th International Symposium on Telecommunications (IST)*, 2016, pp. 139-144, doi: 10.1109/ISTEL.2016.7881798.
- [35] F. A. Balas, O. Almomani, R. M. A. Jazoh, Y. M. Khamayseh and A. Saaidah, "An Enhanced End to End Route Discovery in AODV using Multi-Objectives Genetic Algorithm," *2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)*, 2019, pp. 209-214, doi: 10.1109/JEEIT.2019.8717489.
- [36] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "GreyWolf Optimizer," *Advances in Engineering Software*, vol. 69, pp. 46-61, 2014, doi: 10.1016/j.advengsoft.2013.12.007.
- [37] E. M. Devi and R. C. Suganthe, "Feature selection in intrusion detection grey wolf optimizer," *Asian Journal of Research in Social Sciences and Humanities*, vol. 7, no. 3, pp. 671-682, 2017, doi: 10.5958/2249-7315.2017.00197.6.
- [38] Q. M. Alzubi, M. Anbar, Z. N. M. Alqattan, M. A. Al-Betar, and R. Abdullah, "Intrusion detection system based on a modified binary grey wolf optimization," *Neural computing and applications*, vol. 32, no. 10, pp. 6125-6137, 2019, doi: 10.1007/s00521-019-04103-1.
- [39] Q. Al-Tashi, H. Md Rais, S. J. Abdulkadir, S. Mirjalili and H. Alhussian, "A review of greywolf optimizer-based feature selection methods for classification," in *Evolutionary Machine Learning Techniques*; Springer, Singapore, 2020, pp. 273-286, doi: 10.1007/978-981-32-9990-0\_13.
- [40] M. Madi, F. Jarghon, Y. Fazea, O. Almomani, and A. Saaidah, "Comparative analysis of classification techniques for network fault management," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 28, no. 3, pp. 1442-1457, 2020, doi: 10.3906/elk-1907-84.
- [41] R. A. Khurma, I. Aljarah, A. Sharieh, and S. Mirjalili, "EvoPy-FS: An Open-Source Nature-Inspired Optimization Framework in Python for Feature Selection," *Evolutionary Machine Learning Techniques*, pp. 131-173, 2020, doi: 10.1007/978-981-32-9990-0\_8.
- [42] W. L. Al-Yaseen, "Improving intrusion detection system by developing feature selection model based on firefly algorithm and support vector machine," *IAENG International Journal of Computer Science*, vol. 46, no. 4, pp. 534-540, 2019.
- [43] S. Aljawarneh, M. B. Yassein, and M. Aljundi, "An enhanced J48 classification algorithm for the anomaly intrusion detection systems," *Cluster Computing*, vol. 22, no. 5, pp. 10549-10565, 2019, doi: 10.1007/s10586-017-1109-8.
- [44] A. Taherkhani, G. Cosma, and T. M. Mc Ginnity, "Deep-FS: A feature selection algorithm for deep boltzmann machines," *Neuro computing*, vol. 322, pp. 22-37, Dec. 2018, doi: 10.1016/j.neucom.2018.09.040.
- [45] I. Ahmad, M. Bashari, M. J. Iqbal and A. Rahim, "Performance Comparison of Support Vector Machine, Random Forest, and Extreme Learning Machine for Intrusion Detection," in *IEEE Access*, vol. 6, pp. 33789-33795, 2018, doi: 10.1109/ACCESS.2018.2841987.





- [46] F. Farahnakian and J. Heikkonen, "A deep auto-encoder based approach for intrusion detection system," *2018 20th International Conference on Advanced Communication Technology (ICACT)*, 2018, pp. 178-183, doi: 10.23919/ICACT.2018.8323688.
- [47] C. Yin, Y. Zhu, J. Fei and X. He, "A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks," in *IEEE Access*, vol. 5, pp. 21954-21961, 2017, doi: 10.1109/ACCESS.2017.2762418.
- [48] C. Khammassi and S. Krichen, "A GA-LR wrapper approach for feature selection in network intrusion detection," *Computers & Security*, vol. 70, pp. 255-277, Sep. 2017, doi: 10.1016/j.cose.2017.06.005.
- [49] T. Mehmod and H. B. M. Rais, "Ant colony optimization and feature selection for intrusion detection," in *Advances in Machine Learning and Signal Processing*, New York, NY, USA: Springer, vol. 387, pp. 305-312, 2016, doi: 10.1007/978-3-319-32213-1\_27.
- [50] M. Agarwal, D. Pasumarthi, S. Biswas, and S. Nandi, "Machine learning approach for detection of flooding DoS attacks in 802.11 networks and attacker localization," *International Journal of Machine Learning and Cybernetics*, vol. 7, no. 6, pp. 1035-1051, Dec. 2016, doi: 10.1007/s13042-014-0309-2.
- [51] R. Mathiyalagan and P. V. Eric, "An Efficient Intrusion Detection System Using Improved Bias Based Convolutional Neural Network Classifier," *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, vol. 12, no. 6, Apr. 2021, pp. 2468-2482, doi: 10.17762/turcomat.v12i6.5689.
- [52] R. Mathiyalagan and P. V. Eric, "An Improved Deep Bagging Convolutional Neural Network Classifier for Efficient Intrusion Detection System," *Bulletin of Electrical Engineering and Informatics*, vol. 11, no. 1, Feb. 2022, pp. 405-413, doi: 10.11591/eei.v11i1.3252.
- [53] V. L. L. Thing, "IEEE 802.11 Network Anomaly Detection and Attack Classification: A Deep Learning Approach," *2017 IEEE Wireless Communications and Networking Conference (WCNC)*, 2017, pp. 1-6, doi: 10.1109/WCNC.2017.7925567.
- [54] S. Ding and G. Wang, "Research on intrusion detection technology based on deep learning," *2017 3rd IEEE International Conference on Computer and Communications (ICCC)*, 2017, pp. 1474-1478, doi: 10.1109/CompComm.2017.8322786.

## BIOGRAPHIES OF AUTHORS



**Mathiyalagan Ramasamy**     received the Bachelor of Technology in Information Technology in 2005 from Dr. Navalar Neduncheziyan college of Engineering, Affiliated by Anna University, Tamil Nadu, India, and He received the Master of Engineering in Computer Science and Engineering in 2010 from Oxford college of Engineering, Affiliated by Anna University, Tamil Nadu, India. He is currently working as Assistant Professor in Presidency University, Bengaluru, Karnataka, India. His research interests include intrusion detection system, convolutional neural network, and network security. He can be contacted at email: mathi.prajval@gmail.com.



**Dr. Pamela Vinitha Eric**     Pamela Vinitha Eric is working as a Professor in the Department of Computer Science and Engineering at the Presidency University, Bangalore, India. She received a Doctorate in Philosophy from National Institute of Technology Calicut, India. She has around 25 years of experience in the education sector. Her areas of interest are bioinformatics, data compression, cryptography and network security. She is actively pursuing research and supervises several research scholars. She can be contacted at email: pamela.vinitha@gmail.com.